



**Michigan  
Technological  
University**

College of Computing

Computer Science Department

CS3141 Team Software Project

Spring 2022

## **Laundry Tracker Lite**

Section: R01

Team #: 7

Roll #	Name	Role
30	Colton Sitts	Scrum master
31	Tristan Sorenson	Developer
32	Jacob Suiter	Developer
29	Emilie Rummer	Developer
1	Laura Albrant	Management
5	Ben Cauley	Management

Instructor:

Serein AL-Ratrout

## Abstract

This project is a laundry machine availability tracker intended to give students the ability to see when laundry machines are available both for their convenience and to increase efficient use of the facilities. Students can create an account so that they can check in and out a machine and see which machines are available. It will also notify the students of when their machine is no longer running. Originally input will be handled by the end user, but this can be extended using IoT devices. The main application is hosted on the web which can be accessed cross platform. It can also be extended to collect more sophisticated data to create projections for how long an individual laundry machine will take to complete a cycle, and create usage graphs to suggest optimal times to do laundry throughout the week.

## Table of Contents

<b>Abstract</b>	<b>2</b>
<b>Table of Contents</b>	<b>3</b>
<b>Table of Figures</b>	<b>4</b>
<b>1.1 Introduction</b>	<b>6</b>
<b>1.2 Problem Statement</b>	<b>6</b>
<b>1.3 Aim and Objectives</b>	<b>6</b>
<b>1.4 Stakeholders</b>	<b>7</b>
<b>1.5 Methodology</b>	<b>7</b>
<b>1.6 Tools</b>	<b>8</b>
<b>1.7 High-Level Business Requirements</b>	<b>8</b>
Functional Requirements	8
Non-functional requirements	8
<b>1.8 Product backlog</b>	<b>8</b>
Validation	14
Limitations and Future Work	16
<b>References</b>	<b>20</b>

**Table of Figures**

# Chapter 1

## Specification

## 1.1 Introduction

The purpose of our project is to develop a web application that keeps track of laundry machines all over MTU's campus. It will keep track of when laundry machines are in use and/or unavailable and it will notify people when their laundry is done. We plan on using a Python framework called Flask to build our web app and will use SQL for our database requirements.

## 1.2 Problem Statement

There is currently no way to know which laundry machines are available without physically walking to the laundry rooms. There is also no way to alert someone who has left their laundry in the machine after it is done. This wastes the time and energy of all dorm residents. Many students have busy schedules and do not have time to continuously check if a washer or dryer is available, which can lead to not having an opportunity to do laundry. Another problem is that there is no way to notify people who have left their clothes in a machine for long periods of time after it has finished running. There are a limited number of machines and not only does it take the people who left their clothes longer to do their laundry but it also means other residents do not get an opportunity to use a washer or dryer.

## 1.3 Aim and Objectives

Our aim is to create a laundry tracking web application for students at MTU. Our objectives are to learn a bit of Python and the Flask framework, set up our development environment, allow students to create accounts and check-in and check-out laundry machines, send students notifications for when their laundry is done, all to allow students to more effectively manage their time.

## 1.4 Stakeholders

The stakeholders in this project are MTU students who live in the dorms and use their laundry machines. Another stakeholder is MTU as it will directly affect the way students use their laundry services.

## 1.5 Methodology

We'll begin by researching the development environment we plan on using for the website, making sure to learn as much about Flask as possible. Then we'll move on to designing the overall interface for the website, moving on to properly setting up the database to store the washers and driers, making sure they can be updated and managed as planned; once we've completed that we'll move on to the user side of the website, making sure that they can properly use the site at least at a basic level, and once that's been completed and is relatively bug free, we'll move on to adding more information for the site to gather and more ways that

the user will be able to view the status of any washer they wish to use. After all of the initial plans for the project are completed and it's working properly, we'll move on to touching up the UI and UX; as well as anything else we may think could be beneficial.

## 1.6 Tools

We need to develop both the client and server side of this project. This will require several frameworks used to simplify the process of creating complicated applications that we don't have a lot of initial experience with. We intend to use frameworks to soften the learning curve, here are some examples:

- Python/Flask
- Maybe Javascript
- MySQL (maybe PHP): databases

## 1.7 High-Level Business Requirements

### ● Functional Requirements

- Users create an account
  - Email
  - Password
- Verify the email is a real email
- Have some way to register the laundry machines (labelling in database)
- "Using" a washer is the same as checking it out (max washers checked out per person is 2, max dryers checked out per person is 1)
  - **check out = machine in use**
  - **check in = machine available**
- Notifies user when their laundry is finished
  - And every 15 minutes after if they don't check the machine back in
- Machine check in and check out
  - Auto check in after 12 hours

### ● Non-functional requirements

- Available
- Scalable for more machines

- **Security requirements**

- Lock out the user if an invalid password is entered more than a certain number of times.
- Require a specific level of password sophistication from the user.
- Hash all passwords with a safe hashing algorithm. Never store passwords.
  - SHA-256 hashing using salt. Study how to safely store salt strings.
- Error handling for edge cases.
- SQL injection safety (*needs research*).
- Phishing and Social Engineering protection (*needs research*).

## 1.8 Product backlog

Priority	User Story	Tasks	Estimated effort	Sprint
1	As a developer, I want to set up a development environment so that I can begin to implement the product design.	Research the Flask framework	56 Hours	
		ER Models	56 Hours	
1	As a developer, I want to set up how the site will appear to the user.	Set up a basic wireframe version of the UI	14 Hours	
		Touch up how the site appears to make it look more appealing to the user.	8 Hours	
2	As a user I want to create an account so that I can use the service.	Validate the email.	2 Hours	
		Validate the password.	2 Hours	
		Update the database with name, email, and <i>password hash</i> .	4 Hours	
2	As a developer I want to require a certain level of password sophistication and lock out users after too many password attempts so that I can maintain software security.	Password requirements (length, characters)		
		Password storage (hash, salt)		
		Lock out users after a set number of password attempts.		



2	As a developer I want to create a secure architecture and SQL injection safe software so that users can have guaranteed information safety.	Penetration testing (ongoing)		
2	As an admin, I want to register new laundry machines for users to access. (Backend)	Make it so that the developer can add more laundry machines and locations	1 Hour	
3	As a user, I want to check out a machine so I can do my laundry.	Show available machines.	2 Hours	
		Select a machine to check out.	1 Hour	
		Mark that machine as checked out to the user.	2 Hours	
		Update available machines.	2 Hours	
		UI show remaining time on machines checked out by that user.	2 Hours	
3	As a user, I want to check in a machine after I am done using it so that another user can access it.	Show all checked out machines.	2 Hours	
		Select a machine to check back in.	1 Hour	
		Mark that machine as checked back in.	2 Hours	
		Update available machines.	2 Hours	
4	As a user, I want to see what machines are currently available so that I can do my laundry.	Track available machines.	2 Hours	
		UI shows available machines.	1 Hour	
4	As a user I want to see how much time is left on a machine so that I can do my laundry conveniently.	Track remaining time on each machine.	2 Hours	
		UI show remaining time on all checked out machines.	1 Hour	

# Chapter 2

## Analysis and Design



# Chapter 3

## Implementation

The following [report](#) is a good example that you can follow for implementation please refer to pages 25 - 30, and here is another [example](#) for your reference.

# Chapter 4

## Validation

**For Chapter 4 (Validation):** here you need to write about the process of checking that your software system meets specifications and requirements so that it fulfils its intended purpose, and to confirm or to prove the accuracy of your project.

Write about your testing and validation; **level of testing** you had, unit testing, integration testing, validation testing and acceptance testing. Did you have **manual or automated** testing or both? specify the part(s) that have automated testing and part(s) that have manual testing, and **What is your oracle?**

**Write the test cases** for valid and invalid **input** (please see Week3 Automated Testing/slide 11),

then confirm that no errors in the code and the application is able to operate in required condition (OS, web browsers) and you have created the code correctly.

For validation and acceptance testing write who tested your system? MTU students? computer science student? other department students? your group only? other college students? public users? How many students/users? How many times? could they use it easily or did they make mistakes?

# Chapter 5

## Limitations and Future Work



## For Chapter 5

Limitations: address everything that the project left, if some project backlog items/features/ requirements have not been implemented then mention them in this part with an explanation/justification why you couldn't implement them (Time constraints the time was not enough, some developers were unavailable, because of COVID19, or tool limitation ....etc.). Many students tend to feel that presenting the limits of their work makes work weaker. on the contrary, approaching this section shows maturity for the academic universe, and writing about them actually strengthens your work by identifying any problems before reviewers/readers find them.

Future work : if the limitations can be addressed in the future then add this in here in future work, moreover, if you believe this project can be extendable (add more features/more parts) that the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry as a product so you can give directions for that in future work.

# Chapter 6

## Conclusion

**For Chapter 6 (Conclusion),**: write what you have concluded.

Examples:

I solved many problems in the project...

This application/project/system was applied to improve the learning process.

The results of this project showed that system significantly facilitated the students' learning process.

The system is useless, acceptable, usable, beneficial or maybe enjoyable and why do you believe that

## References

- [1] Dean McPherson, "A Jumbo List of 51 Great Web Development Tools", *Paperform*. Available: <https://paperform.co/blog/web-development-tools/>. 9/19/22
- [2] Jessica Clark, "Top 10 backend frameworks | Which is the best option for you?", *back4app*. Available: [https://blog.back4app.com/backend-frameworks/#ASPNET\\_Core\\_Advantages](https://blog.back4app.com/backend-frameworks/#ASPNET_Core_Advantages). 9/19/22